Center for Open Access in Science ▪ https://www.centerprode.com/ojit.html
Open Journal for Information Technology, 2023, 6(2), 157-172.
ISSN (Online) 2620-0627 ▪ https://doi.org/10.32591/coas.ojit.0602.06157n

A Classifier Model to Detect Phishing Emails Using Ensemble Technique

Fredrick Nthurima & Abraham Matheka

Kenyatta University, Nairobi, KENYA
School of Engineering

Received: 11 September 2023 ▪ Revised: 18 November 2023 ▪ Accepted: 24 December 2023

Abstract

Phishing attacks usually take advantage of weaknesses in the way users behave. An attacker sends an email to the recipient that mimics a genuine email with phishing links. When the recipient clicks on the embedded links, the attacker can harvest critical information like credit card numbers, usernames or passwords as a result of entering the compromised account. Online surveys have put phishing attacks as the leading attack for web content, mostly targeting financial institutions. According to a survey conducted by Ponemon Institute LLC 2017, the loss due to phishing attacks is about $1.5 billion annually. This is a global threat to information security, and it's on the rise due to IoT (Internet of Things) and thus requires a better phishing detection mechanism to mitigate these losses and reputation injury. This research paper explores and reports the use of multiple machine learning models by using an algorithm called Random Forest and using more phishing email features to improve the accuracy of phishing detection and prevention. This project will explore the existing phishing methods, investigate the effect of combining two machine learning algorithms to detect and prevent phishing attacks, design and develop a supervised classifier to detect and prevent phishing emails and test the model with existing data. A dataset consisting of benign and phishing emails will be used to conduct supervised learning by the model. Expected accuracy is 99.9%, with a rate of less than 0.1% for False Negatives (FN) and False Positives (FP).

Keywords: extractive model, abstractive model, hybrid model, natural language processing.

1. Introduction

Phishing attacks usually take advantage of weaknesses in the way users behave. An assailant directs an email to the recipient that mimics a genuine email with phishing links embedded in it. When the recipient clicks on the embedded links, the attacker can harvest critical information like credit card numbers, usernames or passwords as a result of entering the compromised account.

In this chapter, I will introduce the research problem and justification of the problem, what the research will achieve and address, research questions, research scope and the assumptions made during the research work.

Based on the Anti-Phishing Working Group Report 2018, a Phishing attack is the number one attack committed by threat actors as compared to other attacks. It is a form of fraud where the attacker deceives the target for personal gain or reputation damage. Fraud results in

Correspondence: Fredrick Nthurima, Kenyatta University, School of Engineering, Nairobi, KENYA.

users revealing their details like credit card numbers, passwords, PINs, usernames and other sensitive information leading to the compromise of accounts and loss of funds.

Phishing campaigns lure users into giving confidential information by visiting websites that look like legitimate ones (phishing.org, 2018). Phishing is done using a digital gadget like a computer or Ipad through a computer network. Malicious actors usually target the weakest element in the security chain, i.e., end-users (Khonj, Jones & Iraqi, 2013).

With a phishing attack, the attackers package messages so the target users cannot easily detect if the message is not genuine. The users end up clicking on the embedded links, thereby being redirected to the attacker's websites, whereby the attacker can get confidential information like passwords, usernames, credit card numbers etc. This enables threat actors to enter the compromised account and achieve their objectives like data theft, funds transfer or reputation injury.

For instance, a malicious email might have malware which, when clicked by the user, will install itself in the pc or mobile phone and will transfer funds to the account of the attacker whenever the owner of the account tries to transfer cash (Khonji et al., 2013). This attack is called Man in the Browser (MITB), a variant of the Man in the Middle (MITM) attack. The man-in-the-browser attack usually uses vectors like ActiveX components, plugins, or email attachments to deliver the payload to the user's computer or phone.

With the increasing case of cyber-attacks, organizations are looking for safer ways of protecting data and preventing getting hacked or getting hacked again. Design and technology should be greatly improved to prevent hackers from infiltrating networks.

According to (Behdad, French, Bennamoun & Barone, 2012), using better defense systems is not enough to stop malicious actors from penetrating systems since these are sometimes circumvented; a better system should detect malicious activities and prevent them before causing any damage.

1.1 *Problem statement*

Today, many spam email filters exist compared to filters for phishing emails. Many techniques are employed to develop phishing email filters, including Blacklists, Visual similarity, heuristics, and Machine Learning. The results of the above techniques have shown that Machine Learning does offer the best solution for phishing filters (Brown et al., 2017). However, current machine-learning anti-phishing solutions use a single model to detect phishing. According to the results, this could offer better detection accuracy, which currently stands at 98% (Smadi et al., 2015). Moreover, they have used domain/URL characteristics, leaving behind other phishing features in phishing emails and lowering accuracy and detection rates. There is a need to develop a better phishing classifier using a machine learning ensemble model, namely Random Forest, and include other phishing email features to increase detection accuracy. The Random Forest algorithm (RF) employed in this proposal work is a form of a bagging algorithm that categorizes many decision trees (from random training sets) to get improved classification accuracies (Deng et al., 2020).

2. Literature review

This chapter analyzes related works, techniques for text summarization, and various models in use.

2.1 *Understanding phishing attacks*

As per the Counter Phishing Working Gathering (APWG) report, the name "Phishing Movement Patterns Report – fourth Quarter 2017," around 57% of phishing assaults target monetary foundations and settlement administrations. A phishing attack is a very common threat on the internet propagated by malicious actors who lure users into supplying personal information to their websites. By doing so, the malicious actors will be able to harvest critical information about a user ranging from passwords, credit card numbers or usernames, for their malicious objectives.

Researchers have demonstrated that social phishing, where in this case, the word *social* means information related to the target is used, produces very actual results as opposed to regular phishing. Gupta, Prakash, Kompella and Kumar (2015) concluded that if phishing attack emails mimicked a target's ally, the success rate of the phishing attack grew from 16% to 72%. Information's social aspect is valuable to social network operators and attackers. This is made even more possible if the information on social media contains an email address that is genuine or if there is a recent conversation between the target and the mimicked friend.

In the recent past, there has been an emergence with automation of data extraction from social media networks and sites. This has led to the availability of usable data to attackers, which can be used to carry out phishing attacks.

Ofoghi, Ma, Watters and Brown (2017) grouped the following spam attacks; Shared attribute attacks, Relationship-based attacks and Unshared attribute attacks.

With this kind of grouping, Relationship-based attacks use affiliation information only, making this spam attack look like socially engineered phishing which normally tricks users into clicking and inputting sensitive data. With the other attacks, they use information originating from social networks to compromise users and get sensitive data for their malicious actions.

This information originating from social networks is categorized between shared and unshared concerning the target and spoofed friend. Birthday cards can represent unshared information, which looks like genuine cards sent from the target's friend. On the other hand, common attributes like photos where both the victim and mimicked friends are both tagged can be abused for context-aware spam.

Huber, Mulazzani, Leithner, Schrittwieser, Wondracek and Weippl (2011) found that information from various social networks can be abused as a result of weaknesses in the communication channels. This will enable the attackers to acquire sensitive information for their gain. Furthermore, the authors have gone further to demonstrate that the data that is extracted from online networks can be exploited to aim many users with context-ware spam.

Gupta, Prakash, Kompella and Kumar (2015) used a hybrid of two techniques, namely blacklists and heuristics, to detect phishing emails. This hybrid technique attained a False Positive (FP) of 5% and a False Negative (FN) rate of 3%.

Holbrook, Kumaraguru, Downs, Cranor and Sheng (2010) researched several anti-phishing solutions and came up with '*SpoofGuard',* which was designed by Ledesma, Chou, Mitchell and Teraguchi (2014). This solution '*SpoofGuard*' showed an improved detection rate of 38% for False Positives (FP) and 9% for False negatives (FN). Moreover, Nargundkar, Tiruthani and Yu (2017) developed a phishing detection system that used heuristics as a mode of detection. This solution managed to achieve a False positive of 1% and a False Negative of 20%.

Smadi, Aslam, Zhang, Alasem and Hossain (2015) also used the heuristics technique, which achieved a False Positive rate of 3% and 11% False Negative.

Sadeh, Fette and Tomasic (2017) came up with a solution to detect phishing emails by use of Machine Learning. This technique achieved a False Positive rate of 1% and a False Negative

rate of 1.2%. Strobel, Glahn, Moens, De Beer and Bergholz (2010) came up with a hybrid solution by use of machine learning and heuristics, which achieved a False Positive rate of 0.05% and a False Negative rate of 1%.

The above techniques have relatively high False Positives and False Negatives. In our proposed anti-phishing technique, the features are extracted directly from the email, thus eliminating processing overhead and increasing run-time. Thus, by eliminating sending of queries, the proposed model will be faster and remove space complexities.

2.2 *Machine learning anti-phishing methods*

PhishHaven – An efficient real-time AI phishing URLs detection system

This constant computer-based intelligence-produced phishing URL arrangement of recognition was created by Maria Sameen, Kyunghyun Han and Seong Oun Hwang in 2020. This framework utilizes lexical elements-based extraction and investigation techniques. To expand the productivity of the framework, the framework utilizes URL HTML encoding as a lexical element. To detect tiny URLs, the system uses a URL hit mechanism. This system uses an ensemble machine learning model employing the multi-threading approach for the training and testing stages.

The framework utilizes fair democracy to allocate the last marks, i.e., typical or phishing, to the given URLs. This framework accomplished an exactness pace of 98% discovery. The framework involves a worldview execution for troupe AI, which includes equal execution of learning models through multi-stringing. Equal execution in the preparing and testing stages speeds up processes, consequently permitting the location of phishing URLs to progress. The proposed recognition framework flaunts different helpful highlights. First, it is free of any outsider administrations (i.e., WHOIS, Group Cymru, and so on) because every one of the methods, including highlight extraction from a URL assessment and characterization of a URL, is performed inside our location framework. Second, it is free of dialects since it dissects URLs, as it were. Furthermore, third, it can recognize zero-day assaults because the discovery framework dissects URLs in view of the URL's lexical highlights.

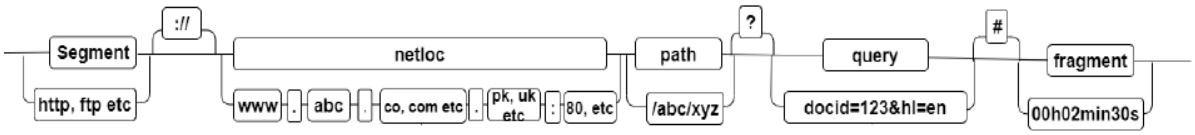


Figure 2.1. Lexical features approach from components of URL

2.3 *Current application of phishing classifiers*

Phishing emails exhibit different features that make them get distinguished from benign emails. These features include; subdomain, prefix_suffix, URL length etc. Mohammad, Thabtah and McCluskey (2013) created unique learning bases using space understanding to detect phishing and legitimate emails. Recent research shows on how to automate the detection of benign and phishing emails. The use of statistical analysis has been used to achieve this, according to Abdelhamid, Thabtah and Ayesh (2014). To study phishing emails better, emails from various sources were grouped based on various phishing features. This grouping was achieved through the recording of occurrences of phishing emails. To improve the detection rate, a larger dataset was collected from various sources (Abdelhamid, Thabtah & Ayesh, 2014).

Several methods have been used to study phishing patterns. These methods are decision tree, support vector machine, Random Forest and Naïve Bayes. A solution called PILFER, which stands for "*Phishing Identification by Learning on Features of Email Received*," was designed to help curb the phishing menace. This solution was used with a case study of 860 phishing emails and 695 benign emails. This experiment was conducted to determine the phishing features in the emails. The features detected by this solution and experiment include IP-based URLs, Email body in HTML format, presence of JavaScript, number of links inside the email and others. Therefore, it was found that PILFER is good at improving the detection of phishing emails by considering the features founds in the emails.

A method called the Random Forest algorithm was used against 2,000 email messages. This experiment aimed to reduce false positives and false negative rates (Akinyelu & Adewumi, 2014). When Random Forest is used with a combination of 15 features, it registers a significant reduction in error rate, becoming the best method in phishing classification and detection hence fitting. Phishing detection models using Random Forest are more dominant concerning detection rate.

Aburrous, Hossain, Dahal and Thabtah (2010) used identified features to classify websites by accurately classifying the identified features. The manual classification was used to group these features into six categories. The categories were then loaded into Waikato Environment for Knowledge Analysis (WEKA) for analysis. This analysis used instances totaling 1006 from PhishTank, whereby four classification algorithms were used to run several experiments. The effectiveness of the features used was measured by classification accuracy. In the experiments using decision tree algorithms, the authors noted a detection rate of 83% of the phishing sites. The authors further pointed out that when this algorithm is coupled with pre-processing, detection accuracy is significantly improved and would be used to make a very good detection model.

A Machine Learning covering algorithm, which goes by the name, Enhanced Dynamic Rule Induction (eDRI), is among the first algorithms to be used as an anti-phishing solution (Thabtah, Qabajeh & Chiclana, 2016). To process the datasets, this Covering algorithm uses frequency and Rule strength as the two major starting points. eDRI only stores "strong" features of the datasets if their frequency exceeds the minimum frequency threshold after scanning all the presented datasets. The stored features are incorporated in the rule, whereas all other values are gotten rid-off in this first process. eDRI removes its training cases, and then strong feature occurrences are updated to signify the inexistence of the instances. This process is done when a rule has been realized. This means eDRI removes its instances and retains strong features. This means eDRI removes features by itself, providing better controllable phishing models. In order to determine eDRI reliability, experiments were carried out on multiple phishing websites. 11,000 websites were collected for these experiments. eDRI showed better results than decision trees and other covering algorithms regarding phishing detection rate. A technique called trial and error Neural Networks which uses Machine Learning, has been condemned due to its time consumption (Mohammad, Thabtah & McCluskey, 2013). For this technique to be effective, a person knowledgeable about the domains is needed during the tuning phase. The elimination of trial and error was proposed but adopted a better self-structuring classification (*Ibid*., 2016). The authors improved the phishing model by improving the learning rate and other parameters and later adding new neurons to the layer that is not visible. This means the features used to build the model are updated during the process of classifier model design.

According to Mohammad, Thabtah and McCluskey (2015), using a dynamic Neural Network model aimed to identify phishing cases from the dataset. Different dataset sizes were used to achieve this, i.e., 100, 200, 500, and 1,000. These experiments showed improved predictions in comparison to Bayesian networks and decision tree techniques.

Since phishing attackers constantly update their phishing techniques, there was a need to develop a more resilient model based on the previous training results (Thabtah, Mohammad & McCluskey, 2014). This aimed to develop a self-learning model to counter the ever-changing techniques used by phishers. The above Neural network algorithm tracks the model's performance by using smart decisions on the results of the validation dataset. The training phase goes as follows; when the error is below the minimum, the algorithm saves up the weights and proceeds with the process. However, if the fault exceeds the lower limit, the algorithm goes further without saving any weights. Parameters can be frequently updated without waiting for the model to be completely built. This experiment revealed that the Neural network model resulted in superior prediction rates compared to traditional techniques like C4.5 and probabilistic.

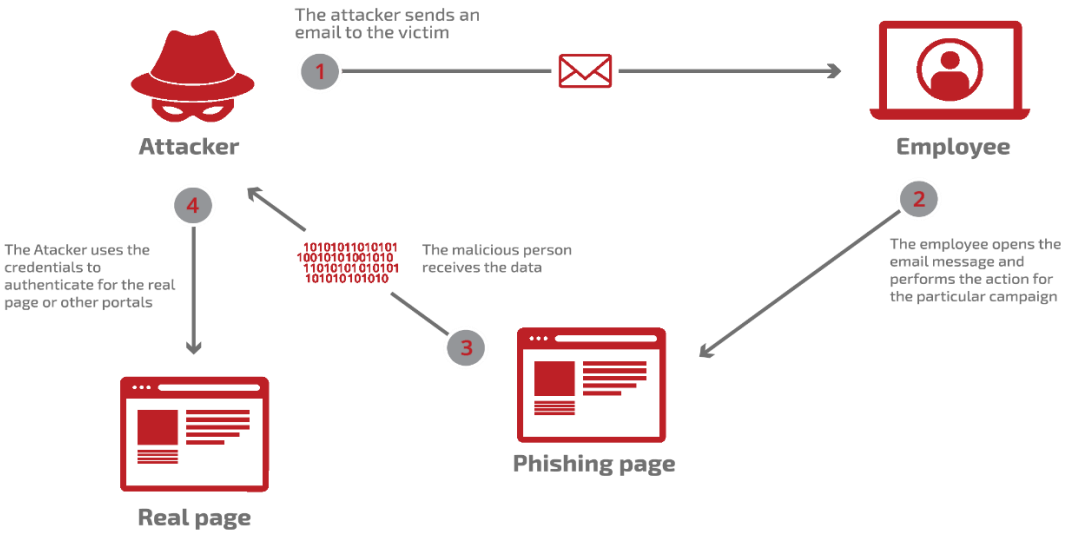


Figure 2.2. Phishing attack execution

2.3.1 *Features used*

This section describes the phishing features that our classifier will use. The features were extracted and identified from the literature and will form a combination of features that effectively classify phishing and benign emails. In this project, we will use 15 features identified from different literature commonly used by phishing attackers.

2.3.2 *IP-based URLs*

Legitimate websites usually have their names on the URL. A case like http://www.mytours.com/ informs the user that someone will visit a website with the domain *mytours.com*. Attackers usually mask their identity by replacing the domain name with an IP address, e.g., http://42.56.100.21/login.asp. By doing this, malicious actors can escape detection by using IP-based URLs, which indicates a possible phishing attack. This discussed feature is identified in the literature (Fette, Sadeh & Tomasic, 2017).

2.3.3 *LINK text mismatch and "HREF" attribute*

A link to another website is usually defined by using an HTML <a> anchor tag. "href" attribute allows a user to visit another website by describing the location of the second website. The content is displayed on the browser when the user clicks the link. This link is in the form of a

href="URL Address"> link text . The link text can be plain text, an image or any element. If there is a match between the link text and the pointed website, the website could be phishing. Two items are checked for mismatch, i.e., link text and href attribute for all the emails. A positive Boolean is recorded when a mismatch is found on these emails.

2.3.4 *Link text of hyperlink*

Phishing emails exhibit certain characteristics on the links that make the emails qualify to be phishing emails. The emails will contain certain words like *click here, log in or update*. Emails are checked for the presence of these words, and a Boolean value is recorded if these words are found or not.

2.3.5 *Dot contained in domain name*

According to Emigh (2016), a legitimate domain name should contain less than three dots. If the number of dots in the URL exceeds three, a binary value of 1 is noted to assist in phishing features.

2.3.6 *HTML email*

MIME standards define every email. MIME standards define what makes up the email and its components. The components are categorized into two types, i.e., *text/plain and text/html*. These are the content-type according to Fette, Sadeh and Tomasic (2017), an email could be a phishing email if it has a "text/html" property. They argued that using HTML links is easier to achieve phishing attacks.

2.3.7 *Use of JavaScript*

JavaScript is a scripting language that is used to perform a particular action. JavaScript is either used in the body of the email using special tags denoted by <script> or can be used on a link using a tag called anchor <a>. Malicious actors make use of JavaScript language to evade detection by hiding information from users with the use of JavaScript. If an email contains a JavaScript code, it is classified as a potential phishing email (Fette, Sadeh & Tomasic, 2017).

2.3.8 *Links found in an email*

The sum of links in an email is registered to detect phishing emails. An email containing many links is a probable candidate for a phishing email. Phishing emails usually have links to external websites that redirect users to the attackers' websites (Yuan & Zhang, 2012).

2.3.9 *Email domain names*

The sum of unique domain characters is extracted for comparison with the referenced URLs. The incidences are recorded, and the value is used as a feature for detecting a phishing email. Each occurring unique domain name is recorded once, and any subsequent occurrence is discarded. It is therefore believed that if an email contains multiple domain names, it is a potential phishing email.

2.3.10 *Body-from domain match*

Domain names form a crucial part of phishing detection. This is because the domain identity of the sender and those in the body of the email should match if an email is to be classified as genuine. A match is performed on the sender's domain name and that of the extracted domain names from the email. The *"From"* field gives the sender's domain name and is compared with our test dataset for a match. If there is a disparity between the comparisons, this suggests it could be a potential phishing email (Altaher, Wan & ALmomani, 2012).

2.3.11 *Word list*

Phishing emails usually contain some occurring words which can be used as phishing detection features. These words will be categorized into six categories, each of which will be used as a single detection feature. This translates to having six different phishing features. Every word is counted in each category, and duplicates are discarded (normalized). These categories are:

a) Confirm; Update

b) Customer; Client; User

c) Restrict, Suspend, Hold

d) Notification, Account, Verify

e) Password, Click, Username, Login

f) Social Security; SSN

2.3.12 *Email datasets, email classifier, email parser, email sanitizer, and email vectorizer ensemble model*

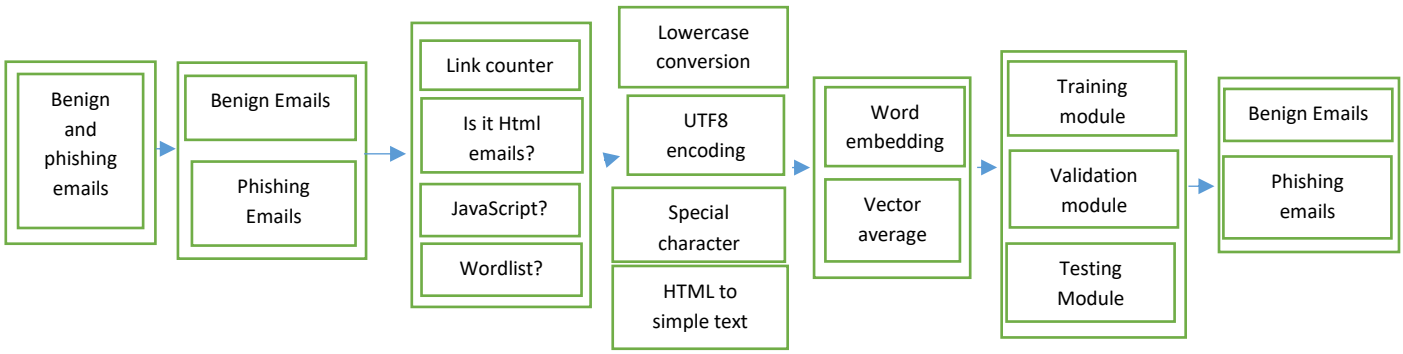


Figure 2.3. Proposed classifier model

3. Methodology

Machine learning is made up of the training phase and the testing phases. We intend to use two datasets to train our model; benign and phishing emails. We will obtain the dataset sets from Alexa for genuine (Benign) emails and PhishTank for luring (phishing) emails. We intend to use a combination of models through the use of an ensemble model called Random Forest (RF) to increase detection accuracy. EMBER (Open-source threat data training set), IBM Watson, Existing Anti-Phishing solutions like Spam Assassin and PhishTank and Alexa for data sets are tools to assist in data modeling. We will use Python libraries like sci-kit-learn, pandas, numpy, matplotlib, and Java.

This research will use a quantitative research method to answer the question of the model's accuracy.

3.1 *Training, testing and validation*

To train and test our classier, we will use a method called 10-fold cross-validation. In this method, the training dataset will be prepared by classifying the dataset into 10 parts. Out of the 10 parts, 9 will be used to train our classifier, and the results obtained from this training will be used to validate the 10th group of the dataset. The process is repeated 10 times so that all ten parts will be used as training and testing data. The cross-checking technique ensures that the information used for training and testing are very different. In Machine Learning projects, this method of 10-fold cross-validation has proven to produce a very good error estimate of the classifier model.

Training the module

Regarding the training module, three constituents are involved: Input Matrix, Target Matrix and Fitness Network. These three components are used consecutively to train the classifier model better and increase the detection rate.

Input Matrix

At this stage, the model uses genuine emails from the Alexa dataset and phishing emails from PhishTank during the training stage of development. The first stage with these email datasets is to *parse* the emails by *email parser*. Then the emails are sanitized by what is known as *email sanitiz*er; lastly, the emails are vectorized by what is known as *email vectorizer*. This research will have x*5 as the logical matrix, indicating 10,000 rows, and the other part of the matrix is 5, meaning 5 columns. 10,000 means there will be a total of 10,000 emails dataset, with 4000 being benign and the other part of 6,000 being known phishing emails.

Every email will have fifteen features with a vector size of 15.

Target matrix

At this stage, the decisions for all benign and phishing emails are found here. The emails stored in the input matrix each produce decisions found in this matrix. In this project, we will have a 10,000*1 matrix meaning that 10,000 will be the total number of emails, whereas l will be vector size. The emails carry 0 or 1, where 0 denotes a benign email while 1 represents a phishing email.

Fitness network

This is where model formalization and testing takes place. The input and target matrix data are utilized in training, formalizing and testing. In this project, 15% will be used for validation, 15% for testing and 70% for training.

3.2 *Model validation and testing*

The validation and testing are the last stage in the model development. At this stage, two matrixes are used: Sample and output.

Sample matrix

This has data from the input matrix, which is usually sample data. After the model is trained, it uses data from the sample matrix, which is used during the testing stage. In our project, this matrix is an m*5 matrix containing sample data from the input matrix.

Output matrix

Data from the sample matrix produces data that is found in this matrix. After training the model, it stores output values in the out matrix. This project represents this by an n * l matrix which contains output data for emails represented in the sample matrix. Using the emails in the sample matrix, the trained model will predict if an email is benign or phishing. The output matrix will store these predictions and will be used to evaluate the performance of the Random Forest algorithm. To achieve our objectives, we plan to use the scikit learn framework to develop, train, validate and then test our classifier model.

The scikit-learn KFold class will automatically implement k-fold cross-validation on the given data set. We intend to use 10-fold cross-validation.

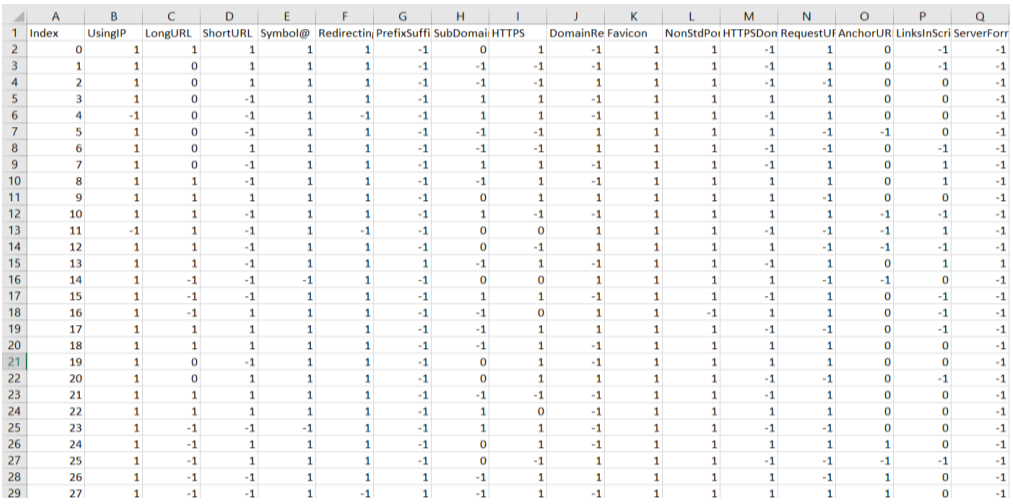
3.3 *Data source*

The experimental data will be collected from two different online sources, whereby one dataset will contain benign URLs while the other will contain phishing URLs. To collect data for the benign URL dataset will be collected from Àlexa, which is a free, open-source data repository site that ranks URLs based on their popularity and non-malicious. The phishing email will be retrieved from the PhishTank website repository. This is a free community website that enables users all over the world to submit, confirm, analyze and share phishing URL data (PhishTank, 2016). The testing datasets will be prepared for testing by cleansing and ensuring no duplicates. This results in clean training and testing datasets. After the dataset preparation, the training dataset will comprise 4,000 URLs, 3,000 from the benign dataset and 1,000 from the malicious set. Moreover, the testing dataset will consist of 6,000 URLs, 2,000 from the benign dataset and 4,000 from the malicious set. To realize the best results, all URLs will be picked randomly, apart from any URLs that will be selected in the testing dataset that don't contain the sets in the training set.

The next stage will be to extract various features from the URLs that have been prepared and cleaned. To realize quality among features, numeral values will be normalized to be between 0 and 1. In this regard, the features are counts and binary representing values of specific entities within the URL.

3.4 *Data set*

The current data set consists of 6,000 emails, with 3,000 of them being phishing emails sourced from Alexa and PhishTank, and 3,000 legitimate emails obtained from the Spam Assassin website (Apacheorg, 2016). This data was collected with the intent of providing a comprehensive overview of the current phishing landscape and offer a basis for further data mining. The Spam Assassin has two different email types: those easily identified as legitimate and those that are hard to differentiate from spam. The hard-to-tell emails, while still legitimate, need a lot extra checking to ensure they are not actually spam.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirectin	PrefixSuffi	SubDomai	HTTPS	DomainRe	Favicon	NonStdPo	HTTPSDon	RequestUI	AnchorUR	LinksInScri	ServerForr
2	0	1	1	1	1	1	-1	0	1	-1	1	1	-1	1	0	-1	-1
3	1	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	-1	-1
4	2	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	0	-1
5	3	1	0	-1	1	1	-1	1	1	-1	1	1	1	1	0	0	-1
6	4	-1	0	-1	1	-1	-1	1	1	-1	1	1	-1	1	0	0	-1
7	5	1	0	-1	1	1	-1	-1	-1	1	1	1	1	-1	-1	0	-1
8	6	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	-1	-1
9	7	1	0	-1	1	1	-1	1	1	-1	1	1	-1	1	0	1	-1
10	8	1	1	-1	1	1	-1	-1	1	-1	1	1	1	1	0	1	-1
11	9	1	1	1	1	1	-1	0	1	1	1	1	1	-1	0	0	-1
12	10	1	1	-1	1	1	-1	1	-1	-1	1	1	1	1	-1	-1	-1
13	11	-1	1	-1	1	-1	-1	0	0	1	1	1	-1	-1	-1	1	-1
14	12	1	1	-1	1	1	-1	0	-1	1	1	1	1	-1	-1	-1	-1
15	13	1	1	-1	1	1	1	-1	1	-1	1	1	-1	1	0	1	1
16	14	1	-1	-1	-1	1	-1	0	0	1	1	1	1	-1	-1	0	-1
17	15	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	1	0	-1	-1
18	16	1	-1	1	1	1	-1	-1	0	1	1	-1	1	1	0	-1	-1
19	17	1	1	1	1	1	-1	-1	1	1	1	1	-1	-1	0	-1	-1
20	18	1	1	1	1	1	-1	-1	1	-1	1	1	1	1	0	0	-1
21	19	1	0	-1	1	1	-1	0	1	-1	1	1	1	1	0	0	-1
22	20	1	0	1	1	1	-1	0	1	1	1	1	-1	-1	0	-1	-1
23	21	1	1	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	0	-1
24	22	1	1	1	1	1	-1	1	0	-1	1	1	1	1	0	0	-1
25	23	1	-1	-1	-1	1	-1	1	1	-1	1	1	-1	-1	0	0	-1
26	24	1	-1	1	1	1	-1	0	1	-1	1	1	1	1	1	0	-1
27	25	1	-1	1	1	1	-1	0	-1	1	1	1	-1	-1	-1	-1	-1
28	26	1	-1	-1	1	1	1	-1	1	1	1	1	1	-1	1	0	-1
29	27	1	-1	-1	1	-1	1	-1	1	-1	1	1	1	1	1	0	-1

Figure 3.3. Sample of the dataset 15 feature

4. Dataset split for training and testing

The evaluation of the performance of the classifying the phishing websites is demonstrated in Figure 4.4 through the use of three distinct machine learning algorithms: Random Forest, Decision Tree classifier and Adaboost. This provides a comprehensive analysis of the Classification Accuracy (CA) of each model in terms of effectiveness and efficiency.

4.1 *Tool*

To evaluate the compatibility of the file, it was converted to a CSV format and tested using the five algorithms selected by the WEKA tool. The results of this experiment will determine whether the file can be used with the WEKA tool or not.

Weka is a powerful set of machine learning algorithms designed to tackle a variety of data mining tasks. It provides a range of tools to pre-process, classify, regress, visualize, and cluster data. These algorithms can be used directly on the dataset or called from Java code, making it ideal for developing new machine learning approaches. With its perplexing capabilities and high burstiness.

Weka is a powerful tool for data mining, offering a broad variety of algorithms to help with any data mining task. This software provides users with the ability to analyze and uncover hidden patterns in large datasets. With Weka, users can quickly and easily explore, visualize, and manipulate data, and ultimately make more informed decisions (Weka, 2016).

4.2 *Experimental results*

Conduct various experiments in different scenarios, evaluate experiments and results using various measures, compare the performance of several experiments, and highlight the results.

The phishing classification model was implemented through generation of the Random Forest Classifier, Decision Tree Classifier and the AdaBoost algorithm. This project aims to evaluate the use of ensemble methods against other algorithms and determine which method is

better for ranking phishing and non-phishing websites. These algorithms are also generated using Python.

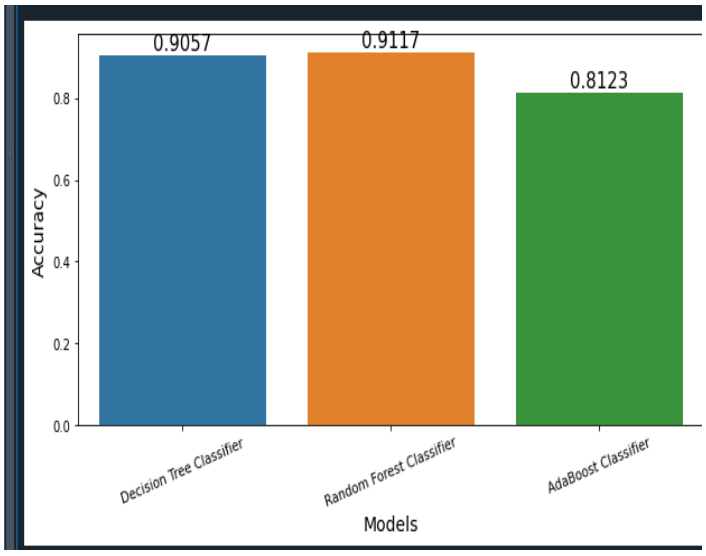


Figure 4.1 Bar plot accuracy of three algorithms

Table 1. Accuracy of three algorithms

No	Algorithm Name	Accuracy
1	Decision Tree	90.59%
2	Random Forest	91.15%
3	Adaboost	81.23%

Figure 4.6 and Table 1 illustrates a difference in algorithms and accuracy. Indicating that the Random Forest classifier gives the biggest accuracy, which is 91.15%, then the Decision Tree, 90.59%, and AdaBoost gives the smallest accuracy, 81.23%. This output demonstrates the accuracy percentage, whereas the training and testing sets are identified with different parameters in the dataset.

5. Conclusion and future work

5.1 *Conclusion*

The results of this project were based on expected results and were instrument tested. This chapter deals with the model's contributions, limitations, suggestions and improvements. Contributions are collected through system goals. The challenges are evaluated by studying the completeness of the model or the challenges and difficulties encountered during the development process of the classifier.

Phishing emails are now a norm in recent years. Phishing is when the victim sends an email requesting key information from the user, which is sent directly to the phisher. Therefore, tracking these emails is necessary. There are numerous innovations to distinguish phishing messages. In any case, they all have restrictions, for example, low exactness, the substance might be like authentic messages and in this manner can't be recognized, and the identification rate should be higher; thus, they have high bogus positives and high misleading negatives. This study evaluated the accuracy of phishing email detection through the use of manual selection feature and also the use of automatic feature selection of three classification algorithms that have high detection rates.

At the end of the process, the two scenarios are compared to determine the method that yields better results in terms of detection rates.

For manual attribute selection, 15 email attributes were chosen and divided into four categories based on email structure (body attributes, header attributes, URL attributes and Java script attributes with external attributes). The results indicate that the body group has the highest accuracy rate in detecting phishing emails, reaching 91.16%.

On the other hand, all but one of the four groups were tested together for accuracy each time.

The results indicate that the highest accuracy rate, 98.25, is achieved if the URL attribute group is removed from all the attributes.

Using auto-selection of the project testing, the accuracy was tested on three sets of auto-selected features, which are generated by the system. The results showed a deviation in accuracy between the three categories, with the highest group being the third one achieving 98% precision.

5.2 *Future work*

More work is needed for future feature selection techniques since selection techniques still need to be refined to cope with new techniques that anglers develop over time. Thusly, we propose to obtain another mechanized device to separate new elements from new crude messages to improve phishing email location precision and adapt to the extension of phishing methods.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public commercial, or not-for-profit sectors.

The authors declare no competing interests.

References

Abdelhamid, N., & Thabtah, F. (2014). Associative classification approaches: Review and comparison. *Journal of Information and Knowledge Management* (JIKM), *13*(3).

Aburrous, M., Hossain, M., Dahal, K. P., & Thabtah, F. (2010). Experimental case studies for investigating e-banking phishing techniques and attack strategies. *Journal of Cognitive Computation*, *2*(3), 242-253.

Afroz, S., & Greenstadt, R. (2011). PhishZoo: Detecting phishing websites by looking at them. In *Fifth International Conference on Semantic Computing* (18-21 September). Palo Alto, California USA, 2011. IEEE.

Akinyelu, A. A., & Adewumi, A. O. (2014). Classification of phishing emails using random forest machine learning technique. *Journal of Applied Mathematics*, vol. 2014, Article ID 425731, 6 pages, 2014.

Altaher, A., Wan, T. C., & ALmomani, A., (2012). Evolving fuzzy neural network for phishing emails detection. *Journal of Computer Science*, *8*(7).

APWG Phishing Attack Trends Reports (2018). https://www.antiphishing.org/resources/apwg-reports/.

Basnet, R., Mukkamala, S., & Sung, A. H. (2008). *Detection of phishing attacks: A machine learning approach*. Soft Computing Applications Industry, pp. 373-383.

Bayesian network classifiers in Weka (2004). Working paper series. University of Waikato, Department of Computer Science. No. 14/2004. Hamilton, New Zealand: University of Waikato.

Behdad, M., French, T., Bennamoun, T., & Barone, L. (2012). *Nature-inspired techniques in the context of fraud detection*. IEEE Transactions on Systems, Man, and Cybernetics C.

Bouckaert, R. (2004). *Bayesian network classifiers in Weka* (Working paper series. University of Waikato, Department of Computer Science. No. 14/2004). Hamilton, New Zealand: University.

Brown, S., Ofoghi, B., Ma, L., & Watters, P. (2017). Detecting phishing emails using hybrid features. *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing* (UIC-ATC '17), IEEE, Australia.

Cranor, L. F., J. I. Hong, & Y. Zhang (2016). Cantina: A content-based approach to detecting phishing websites. In *16th International World Wide Web Conference* (WWW '07), Canada.

Cutler, A., & Breiman, L. (2007). *Random forests-classification description*. Department of Statistics Homepage.

Emigh, A. (2016). Phishing attacks: information flow and chokepoints. In *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. USA.

Fette, I., Sadeh, N., & Tomasic, A. (2017). *Learning to detect phishing emails. Proceedings of the 16th international conference on the World Wide Web*. 649-656.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119-139.

Gaines, B. R., & Compton, J. P. (1995). Induction of ripple-down rules applied to modeling large databases. *Intell. Inf. Syst.*, *5*(3), 211-228.

Gupta, M., Prakash, P., Kompella, R. R., & Kumar, M. (2015). PhishNet: Predictive blacklisting to detect phishing attacks. *IEEE Conference on Computer Communications*.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA Data Mining Software: An Update; *SIGKDD Explorations*, Volume 11, Issue 1.

Han, W., Cao, Y., & Le, Y. (2015). Anti-phishing based on automated individual white-list. *4th ACM workshop on digital identity management (DIM)* (pp. 51-59). ACM USA.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*, 63-90.

Huber, M., Mulazzani, M., Leithner, M., Schrittwieser, S., Wondracek, G., & Weippl, E. (2011). Computer security applications. 27th Annual Computer Security Applications Conference.

Khonji, M, Jones, A., & Iraqi, Y. (2013). *Phishing detection: A literature survey*. IEEE Communications & Surveys Tutorials.

Ledesma, R., Chou, N., Mitchell, J. C., & Teraguchi, Y. (2014). Client-side defense against web-based identity theft. *11th Annual Network & Distributed System Security Symposium*. USA.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill, New York, NY, USA.

Mohammad, R., Thabtah, F., & McCluskey L. (2015B). *Phishing websites dataset*. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites. Accessed January 2016.

Mohammad, R., Thabtah F., & McCluskey L. (2014A). Predicting phishing websites based on self-structuring neural network. *Journal of Neural Computing and Applications*, *25*(2), 443-458. ISSN 0941-0643. Springer.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2013). Predicting phishing websites using neural network trained with back-propagation. Las Vegas, *World Congress in Computer Science, Computer Engineering, and Applied Computing*, pp. 682-686.

Nargundkar, S., Tiruthani, N., & Yu, W. D. (2017). PhishCatch – A phishing detection tool. *33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC '17)*, USA.

Nazif, M., Ryner, B., & Whittaker, C. (2010). Large-scale automatic classification of phishing pages. *17th Annual Network & Distributed System Security Symposium (NDSS '10)*. The Internet Society, USA.

Platt, J. (1998). *Fast training of SVM using sequential optimization: Advances in kernel methods support vector learning*. MIT Press, Cambridge, 1998, pp. 185-208

Qabajeh I., Thabtah, F., & Chiclana, F. (2015). Dynamic classification rules data mining method. *Journal of Management Analytics*, *2*(3), 233-253.

Quinlan, J. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Sadeh, N., Fette, I., & Tomasic, A. (2017). Learning to detect phishing emails. *16th International World Wide Web Conference (WWW '17)*. Canada.

Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., & Downs, J. (2010). Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions. *Proceedings of the 28th international conference on human factors in computing systems - CHI* '10, 373–382. https://doi.org/10.1145/1753326.1753383

Smadi, S., Aslam, N., Zhang, L., Alasem, R., & Hossain, M. A. (2015). Detection of phishing emails using data mining algorithms. *9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*.

Strobel, S., Glahn, S., Moens, M. F., & Bergholz, A. (2010). New filtering approaches for phishing email. *Journal of Computer Security*, 18(1), 7-35.

Sung, A. H., Basnet, R., & Mukkamala, S. (2008). Detection of phishing attacks: A machine learning approach. In *Soft Computing Applications in Industry*. Germany.

Tan, C. L., Chiew, K. L., & Sze, S. N. (2017). Phishing webpage detection using weighted URL tokens for identity keywords retrieval. In Ibrahim, H., Iqbal, S., Teoh. S., & Mustaffa, M. (Eds). *9th*

International conference on Robotic, Vision, Signal Processing and Power Applications. *Lecture Notes in Electrical Engineering*. Vol. 398. Springer, Singapore.

Thabtah, F., Mohammad, R., & McCluskey, L. (2016B). A dynamic self-structuring neural network model to combat phishing. In *Proceedings of the 2016 IEEE World Congress on Computational Intelligence*. Vancouver, Canada.

Thabtah, F., Qabajeh, I.., & Chiclana, F. (2016A). Constrained dynamic rule induction learning. *Expert Systems with Applications*, *63*, 74-85.

Wattenhofer, R., Burri, N., & Albrecht, K. (2015). Spamato-an extendable spam filter system. In *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS '15)*. USA.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. PubMed Central.

Yuan, Y., & Zhang, N. (2012). *Phishing detection using neural network*. http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf

Zhang, Y., Cranor, L. F., Hong, J. I, & Egelman, S. (2016). Finding phish: an evaluation of anti-phishing toolbars. *14th Annual Network & Distributed System Security Symposium*. USA.